Computing Beyond Moore’s Law

John Shalf

Department Head for Computer Science

Lawrence Berkeley National Laboratory

July 7, 2022

The Future of Computing Beyond Moore's Law

That is actually what I will be talking about today…

Technology Scaling Trends

Exascale in 2021… and then what?

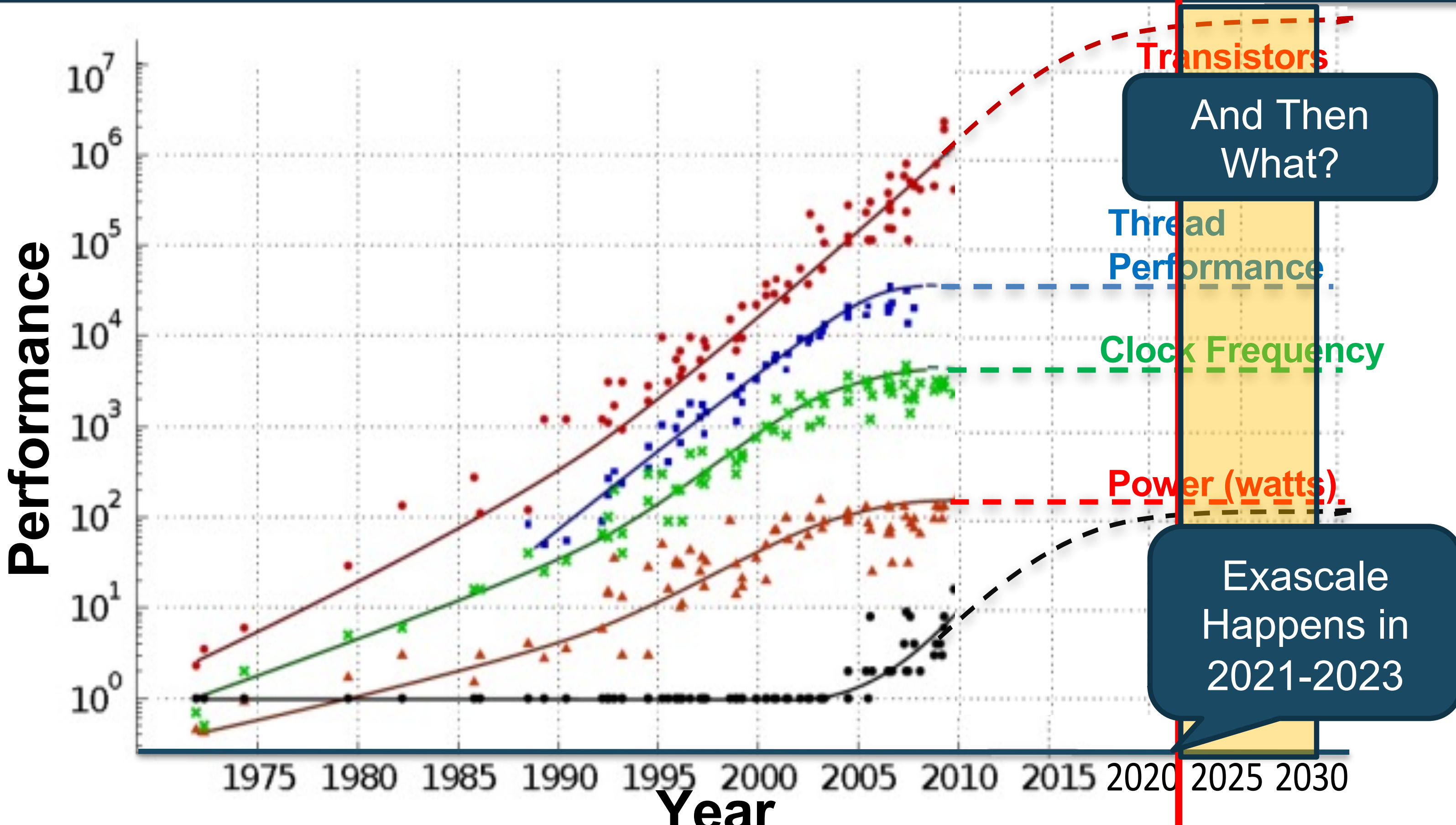
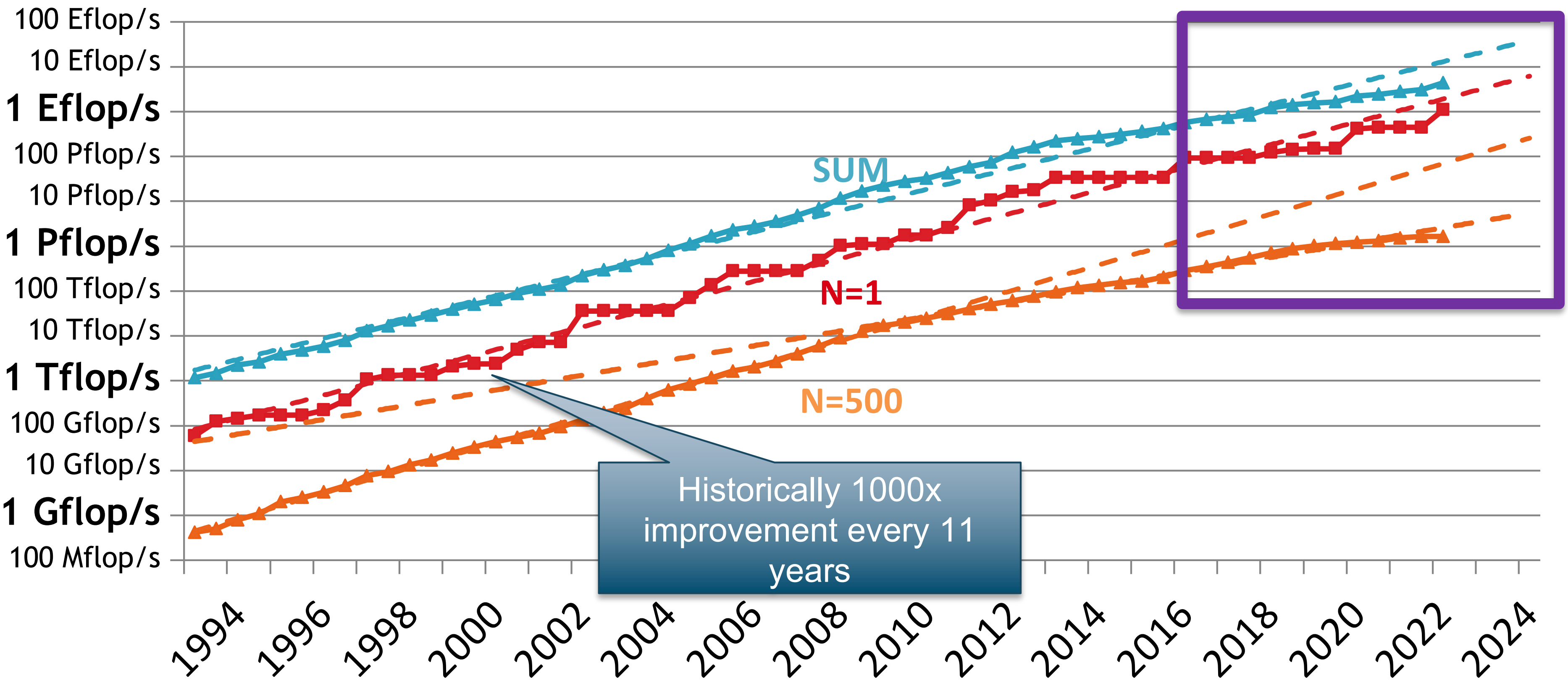
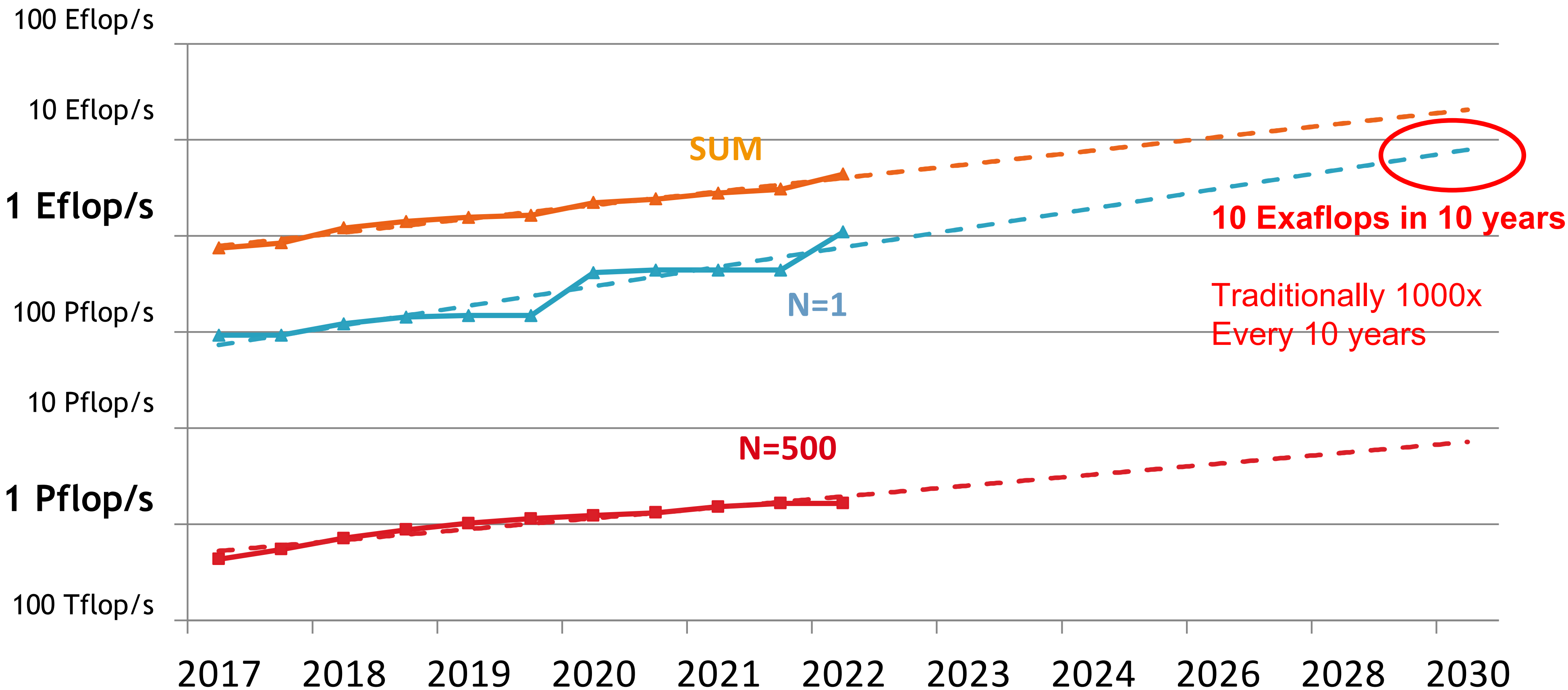


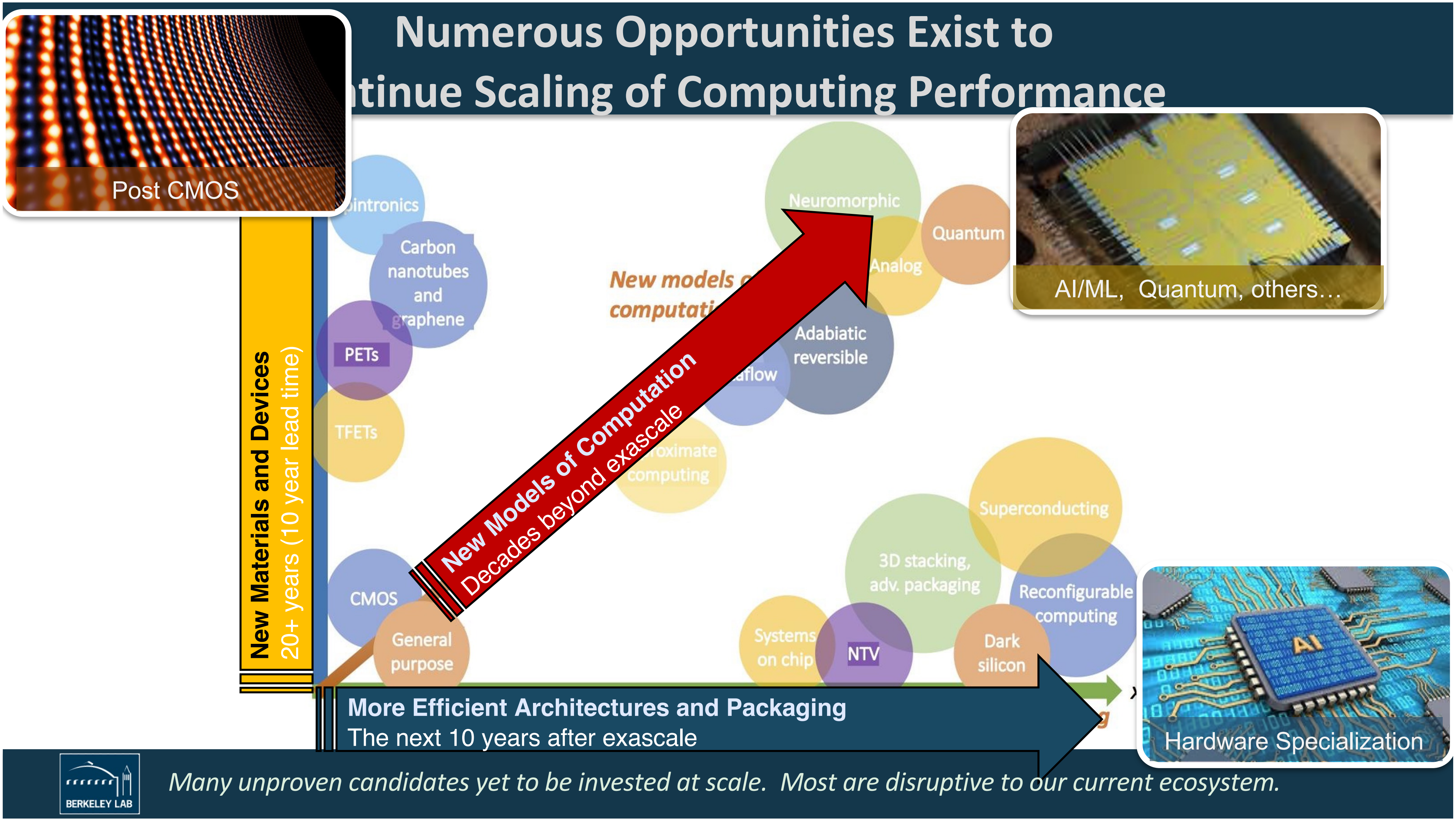
Figure courtesy of Kunle Olukotun, Lance Hammond, Herb Sutter, and Burton Smith

Projected Performance Development



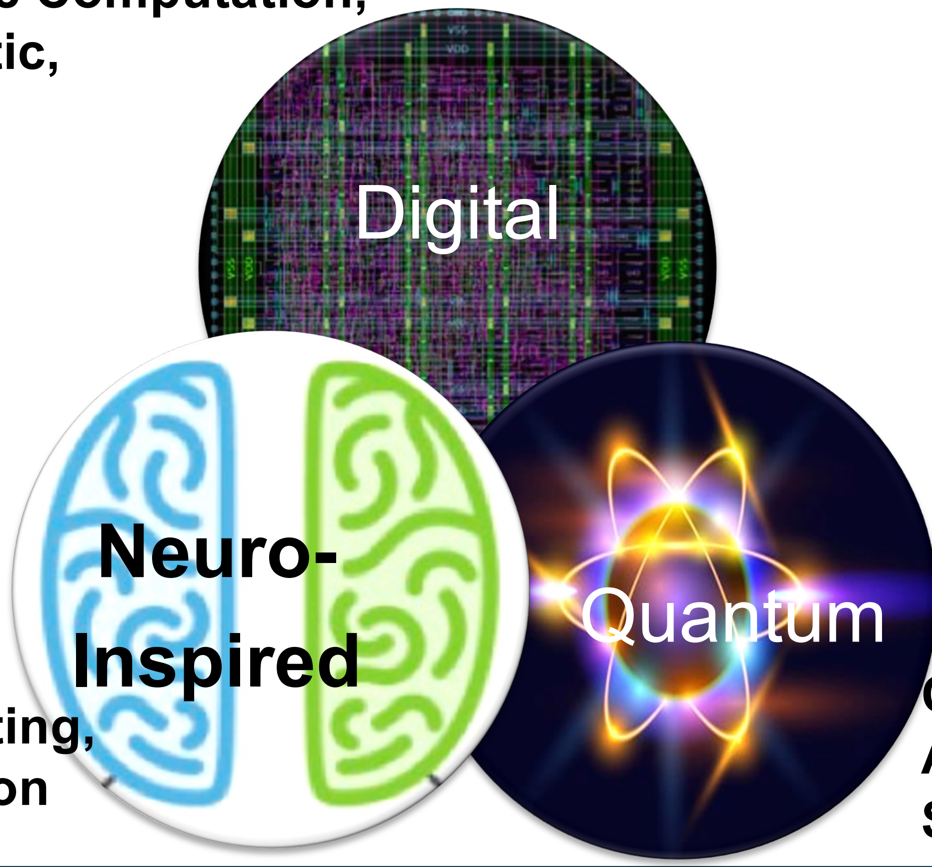
PROJECTED PERFORMANCE DEVELOPMENT




Numerous Opportunities Exist to
ntinue Scaling of Computing Performance
Post CMOS
AI/ML, Quantum, others…
Hardware Specialization
New Materials and Devices
20+ years (10 year lead time)
New Models of Computation
Decades beyond exascale
More Efficient Architectures and Packaging
The next 10 years after exascale
pintronics
Carbon nanotubes and graphene
PETs
TFETs
CMOS
General purpose
New models o
computati
Neuromorphic
Quantum
Analog
Adabiatic reversible
aflow
oximate computing
Superconducting
3D stacking, adv. packaging
Reconfigurable computing
Systems on chip
NTV
Dark silicon
BERKELEY LAB
Many unproven candidates yet to be invested at scale. Most are disruptive to our current ecosystem.

Beyond Moore Computing Taxonomy

Symbolic Computation,
Arithmetic,
Logic



Cognitive Computing,
Pattern Recognition

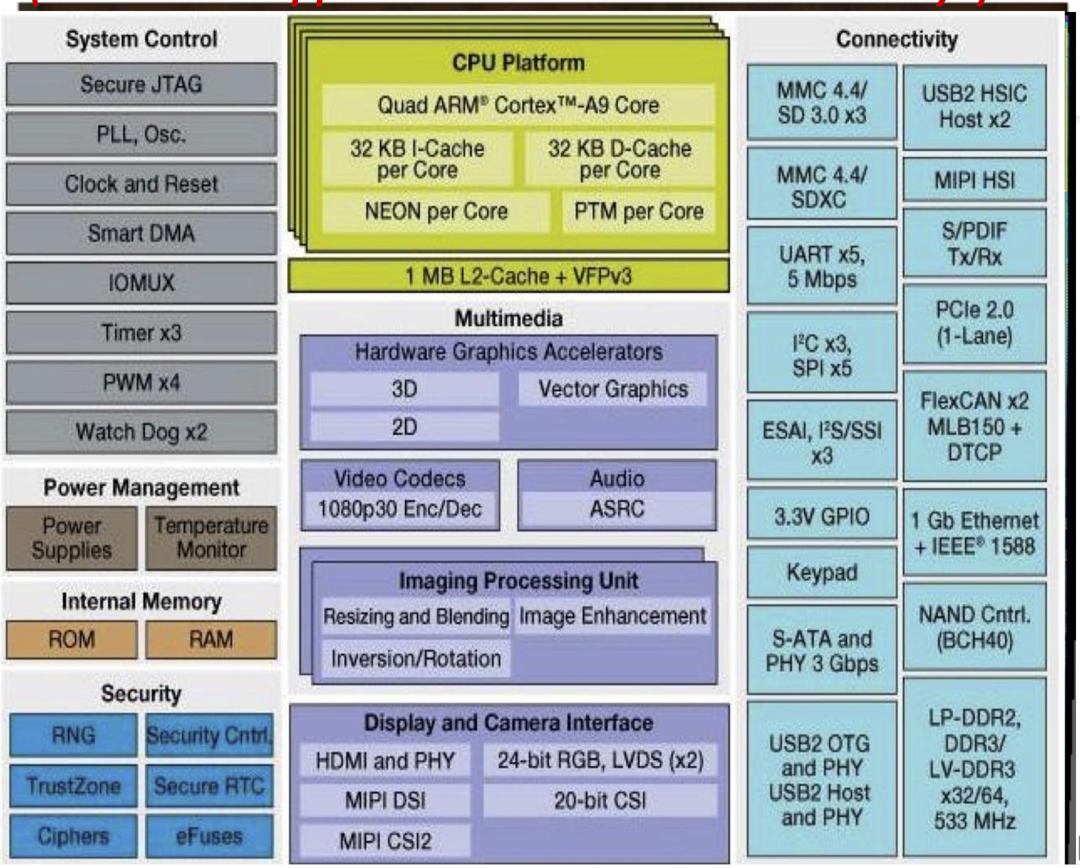
Combinatorial/NP,
Annealing/Optimization,
Simulated Atoms

Extreme Hardware Specialization is Happening Now!

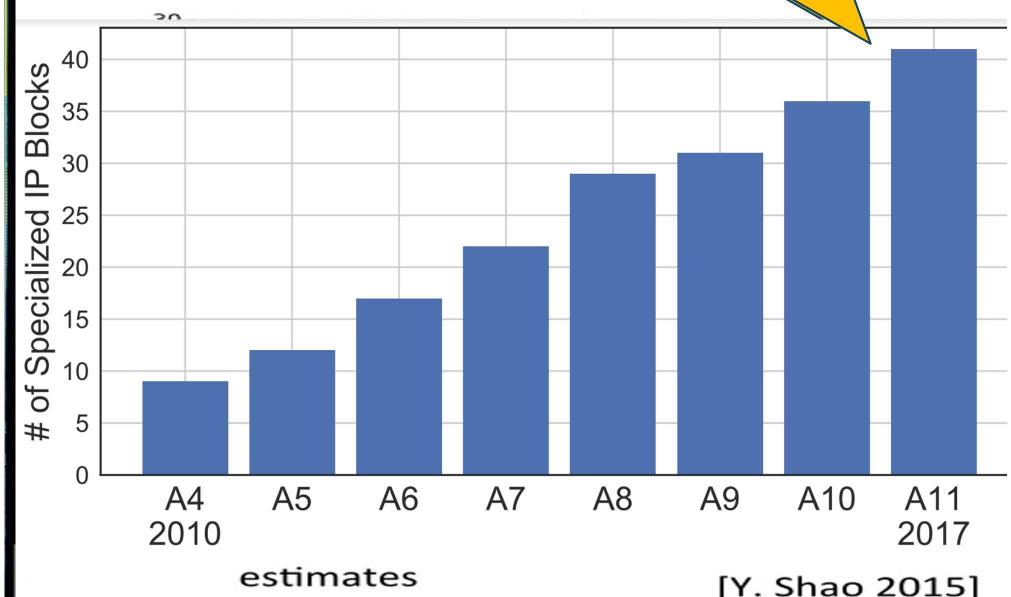
This trend is already well underway in broader electronics industry

Cell phones and even megadatacenters (Google TPU, Microsoft FPGAs…)

(and it will happen to HPC too… will we be ready?)



40+ different heterogeneous accelerators in Apple A11 (2019)

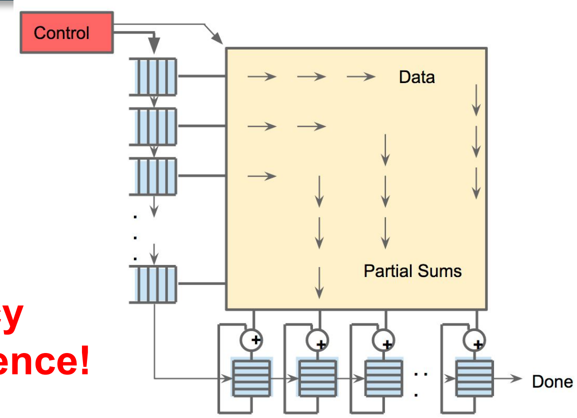


[www.anandtech.com/show/8562/chipworks-a8]

Large Scale Datacenters also Moving to Specialized Acceleration
The Google TPU

Deployed in Google datacenters since 2015

- **"Purpose Built" actually works** - *Only hard to use if accelerators was designed for something else*
- **Could we use TPU-like ideas for HPC?**
- **Specialization will be necessary to meet energy-efficiency and performance requirements for the future of DOE science!**



of the Matrix Multiply Unit. Software
B input is read at once, and they instantly
f 256 accumulator RAMs.

Model	MHz	Measured Watts		TOPS/s		GOPS/s /Watt		GB/s	On-Chip Memory
		Idle	Busy	8b	FP	8b	FP		
Haswell	2300	41	145	2.6	1.3	18	9	51	51 MiB
NVIDIA K80	560	24	98	--	2.8		29	160	8 MiB
TPU	700	28	40	92	--	$\boxed{2,300}$		34	28 MiB

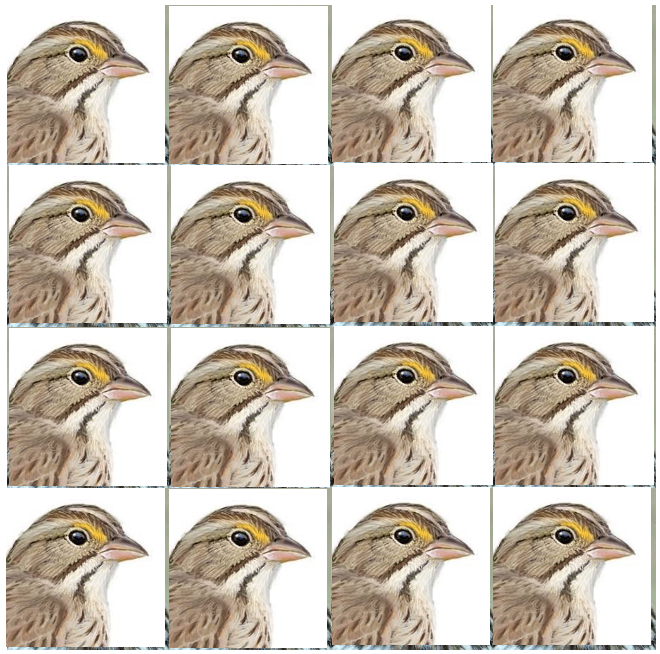
Notional exascale system:
2,300 GOPS/W →? 288 GF/W (dp) → a 3.5 MW Exaflop system!

Specialization:
Natures way of Extracting More Performance in Resource Limited Environment

Powerful General Purpose

Xeon, Power

Many Lighter Weight
(post-Dennard scarcity)



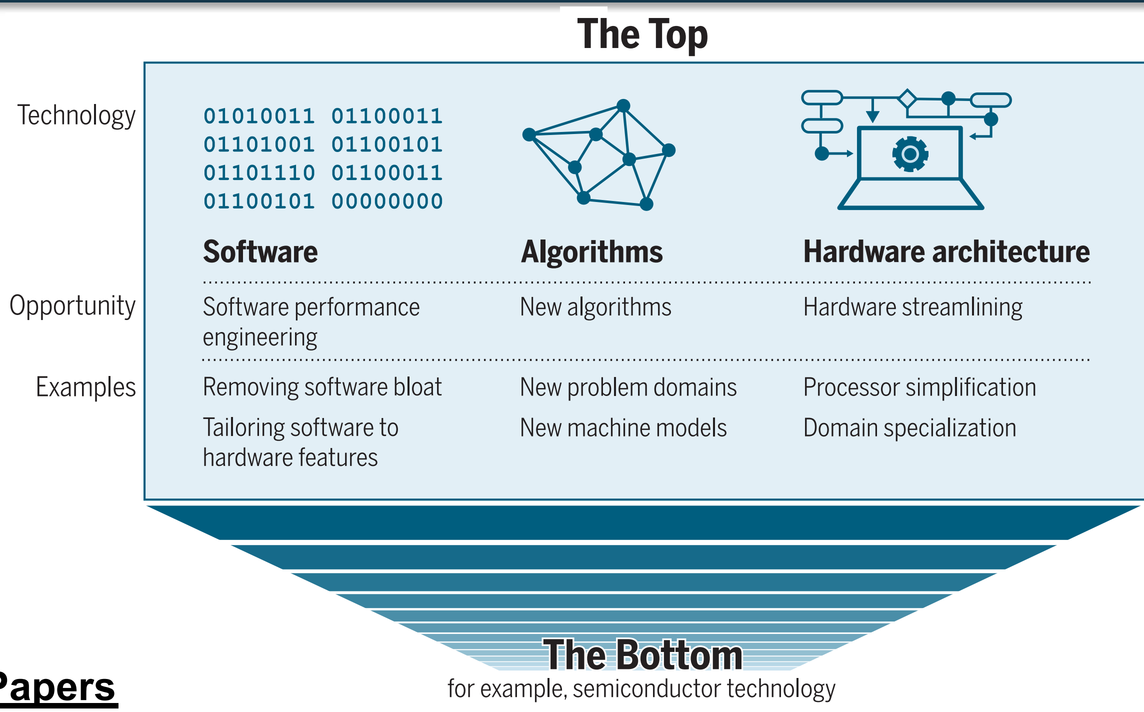
KNL AMD, Cavium/Marvell, GPU

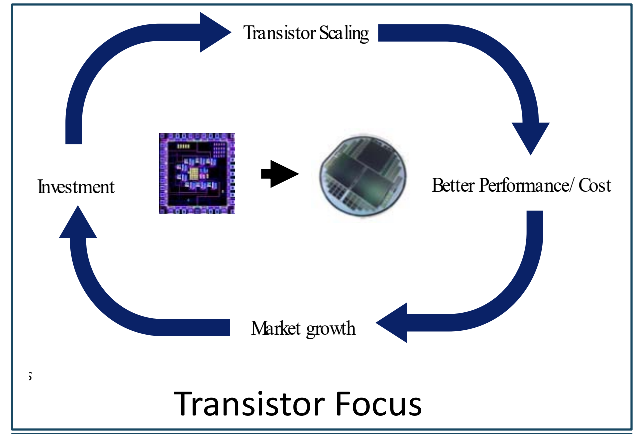
Many Different Specialized
(Post-Moore Scarcity)

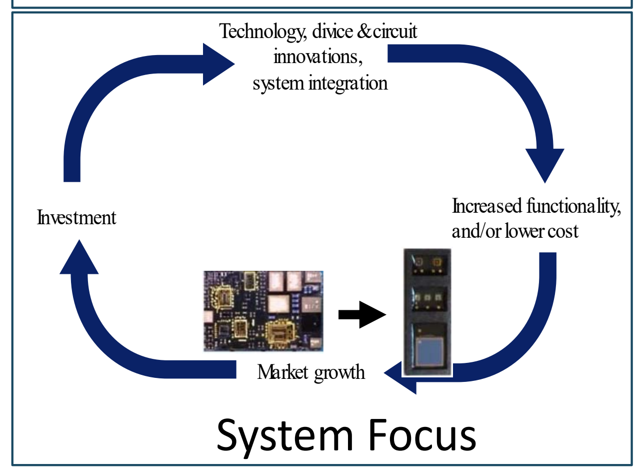
Apple, Google, Amazon

Neil Thompson: Economics of Post-Moore Electronics

http://neil-t.com, MIT CSAIL, MIT Sloan School



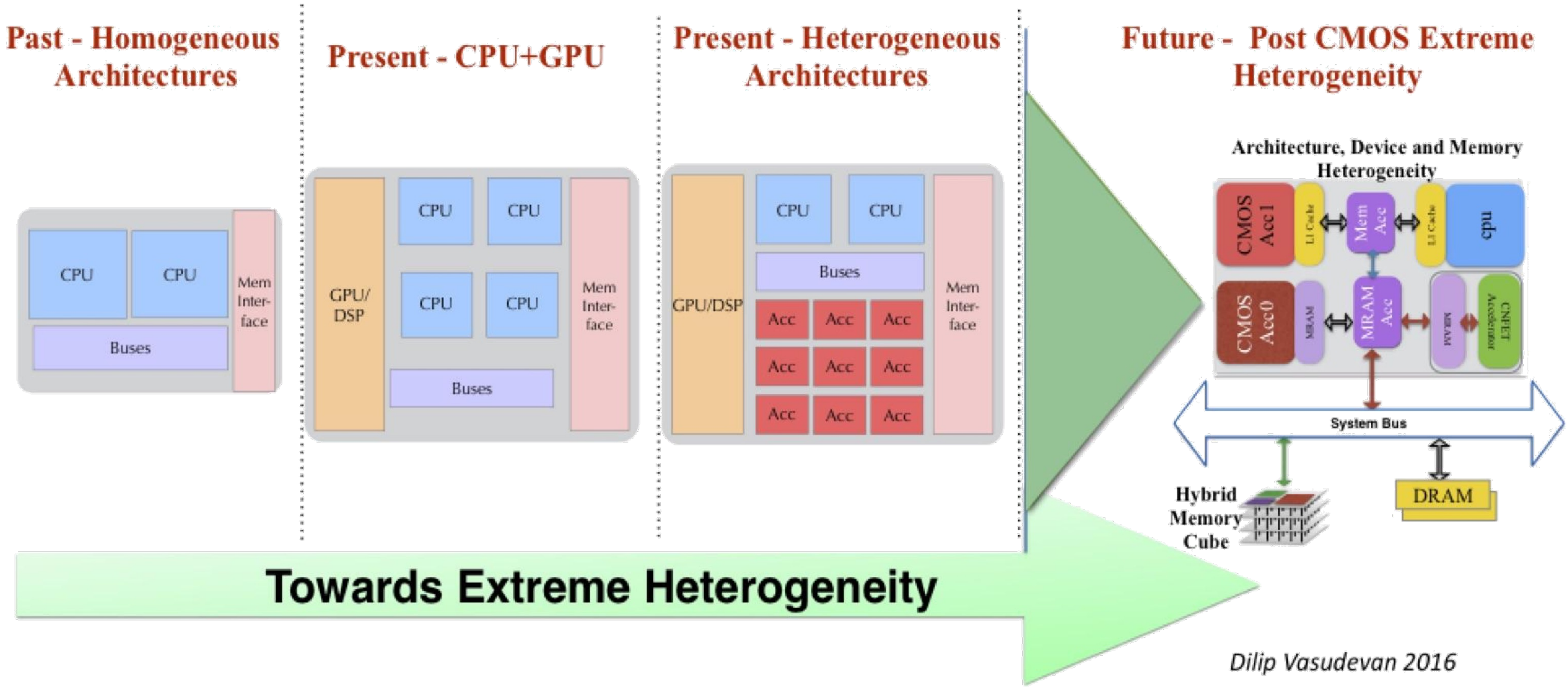




Papers

1. The Economic Impact of Moore's Law
2. There's Plenty of Room at the Top: What will drive computer performance after Moore's Law?
3. The Decline of Computers as a General Purpose Technology

The Future Direction for Post-Exascale Computing



Dilip Vasudevan 2016

Industry: Heterogeneous Integration Roadmap

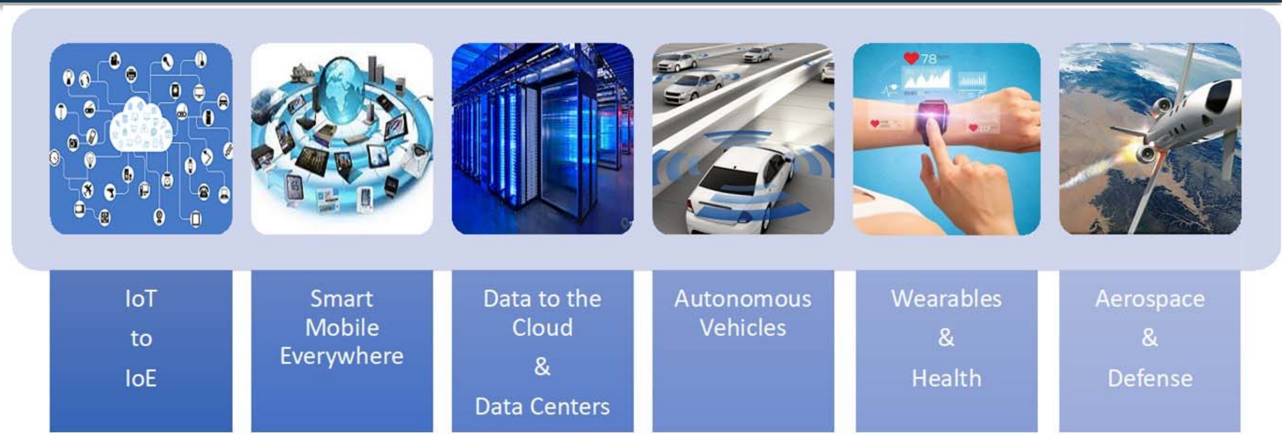
HIR

HETEROGENEOUS INTEGRATION ROADMAP

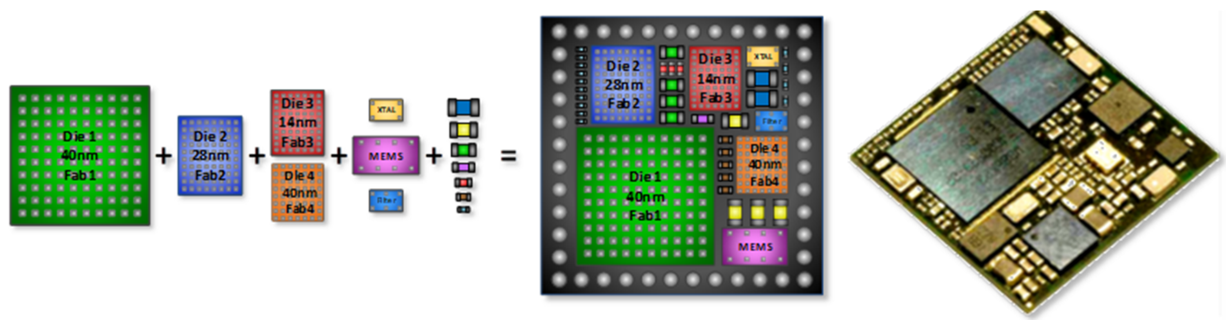
2019 Edition

http://eps.ieee.org/hir

HPC and Megadatacenters is 2nd chapter



All future applications will be further transformed through the power of AI, VR, and AR.

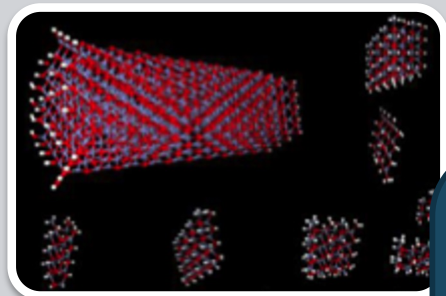
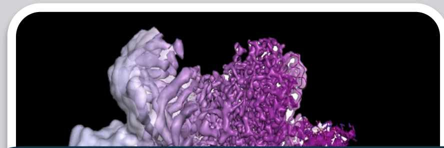


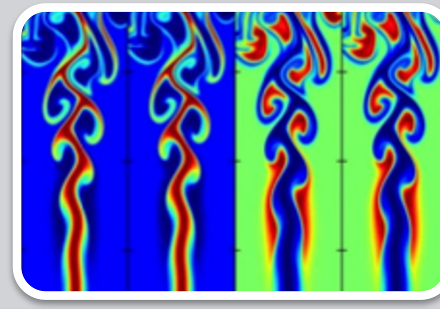
Die + Heterogeneous

System in Package (SiP)

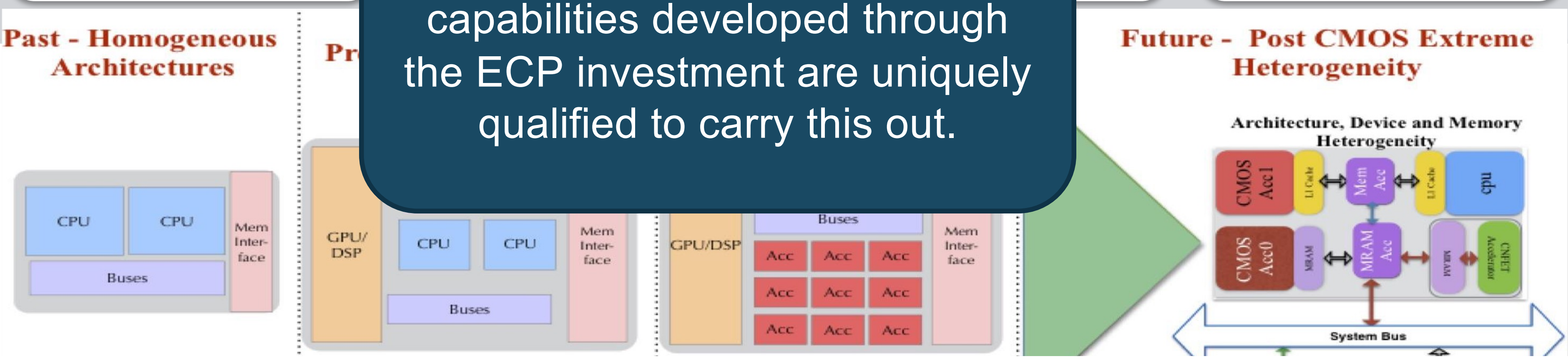
Architecture Specialization for Science

(hardware is design around the algorithms) can’t design effective hardware without applied math



The multi-disciplinary codesign capabilities developed through the ECP investment are uniquely qualified to carry this out.



This needs to be done in close collaboration with applied mathematics

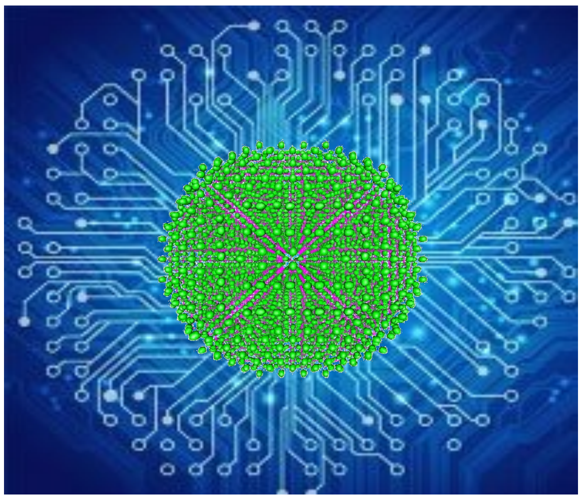
You cannot specialize effectively without deep understanding of the algorithmic target for those specializations

Need to know degrees of freedom for reformulating the mathematics to match hardware strengths

Potential Paths Forward for HPC

1. **Specialization**: purpose built machines for big science targets
2. **Heterogeneity**: Co-integration of many heterogeneous accelerators
3. **Disaggregation**: Photonic MCMs to enable reconfigurable systems

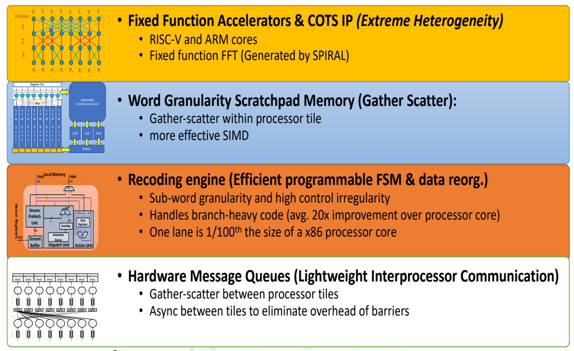
Post Exascale: Heterogeneous Computing Research Directions



Specialization

Purpose built machines for big science targets.

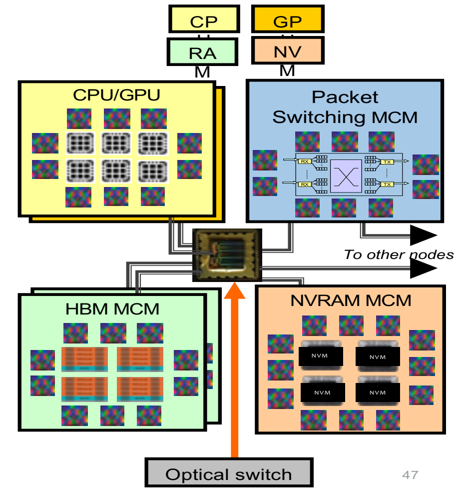
Example: Google TPU. For DOE, DFT is 25% of workload



Heterogeneous Integration

Co-integration of many heterogeneous accelerators

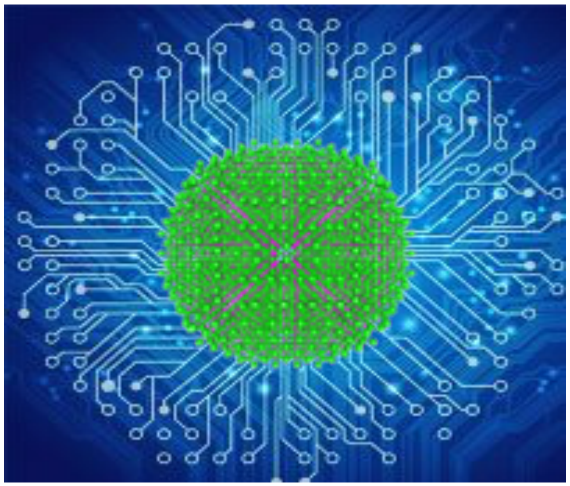
Example: Apple Bionic chip, AWS Graviton2, Project38.



Resource Disaggregation

Photonic MCMs to enable reconfigurable nodes/systems

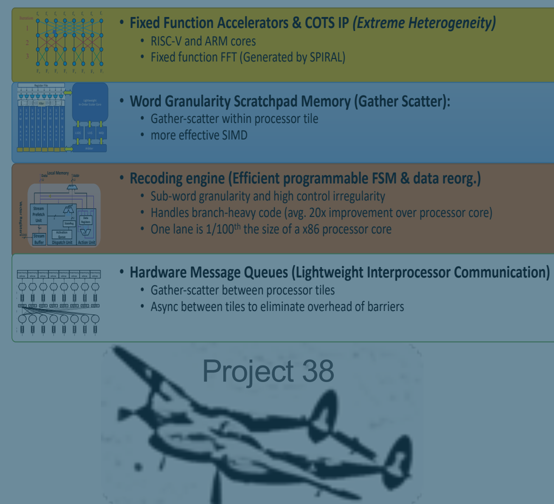
Example: Facebook/Google. Just DRAM utilization diversity in DOE could benefit from this.



Specialization

Purpose built machines for big science targets.

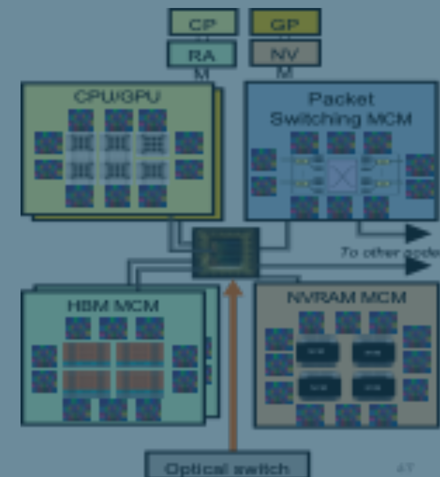
Example: Google TPU. For DOE, DFT is 25% of workload



Heterogeneous Integration

Co-integration of many heterogeneous accelerators

Example: Apple Bionic chip, AWS Graviton2, Project38.



Resource Disaggregation

Photonic MCMs to enable reconfigurable nodes/systems

Example: Facebook/Google. Just DRAM utilization diversity in DOE could benefit from this.

Algorithm-Driven Design of Programmable Hardware Accelerators

Example: LS3DF/Density Functional Theory (DFT)

What: Design the hardware acceleration around the target algorithm/application

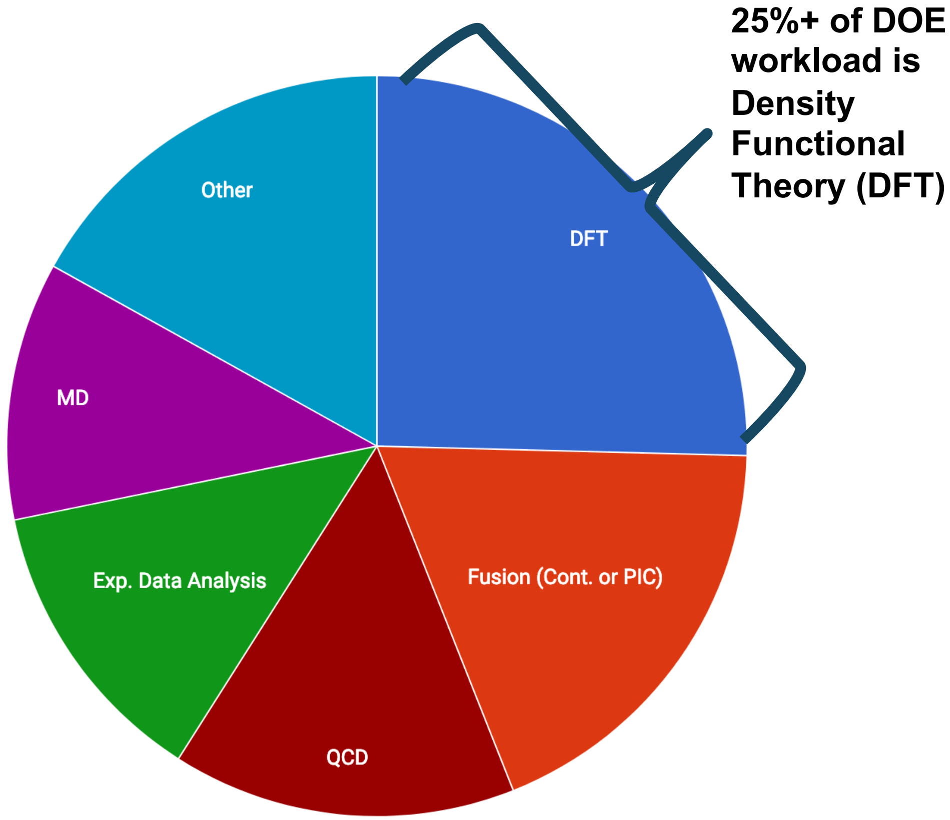
– Purpose-built acceleration
– Science-led reference algorithm design

Why: Huge opportunities to improve performance density and efficiency

– FFT hardware accelerator 50x-100x faster than GPU (using SPIRAL generator)

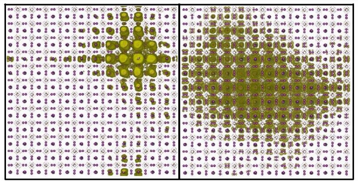
How: Target Density Functional Theory

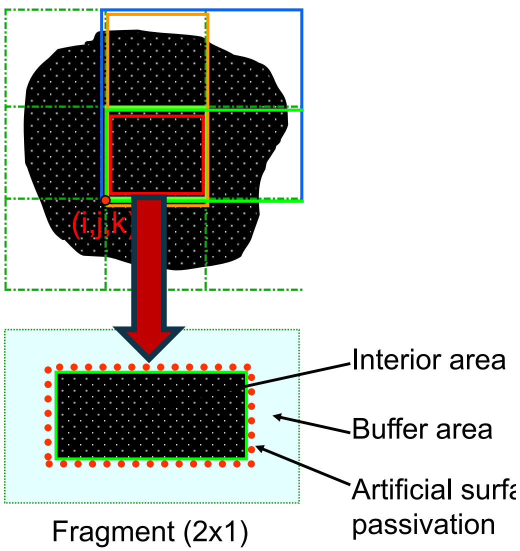
1. Large fraction of the DOE workload
2. Mature code base and algorithm
3. LS3DF formulation minimizes off-chip communication and scales O(N)



The DFT kernel for each fragment

Communication Avoiding LS3DF Formulation – Scales O(N)





$$h(i,j) = \langle \psi_i | H | \psi_j \rangle$$

$$h(i,j) = \langle \psi_i | H | \psi_j \rangle$$

- Sub_diag, *
- Hpsi, *
- Precond. CG step
- Projection, *
- Line minimiz.
- Orth., *
- Sub_diag, *

O(N² Log(N))

Comm bound if non-local

3D parallel FFT

TSQR & Choelesky
ZGEMM

O(N³)
Compute-bound

LS3DF O(N) Algorithm Formulation
Minimizes off-chip Communication

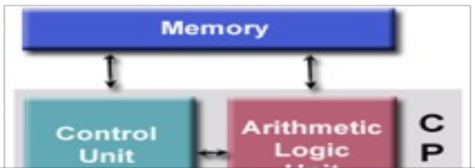
One patch per CGRA
400 bands/patch

Compute Intensive Kernels
Targeted for HW Specialization

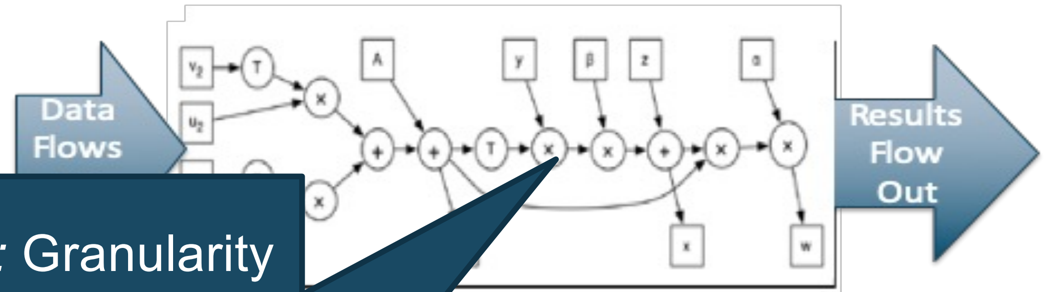
Von-Neumann Instruction Processors vs. Hardware Circuits

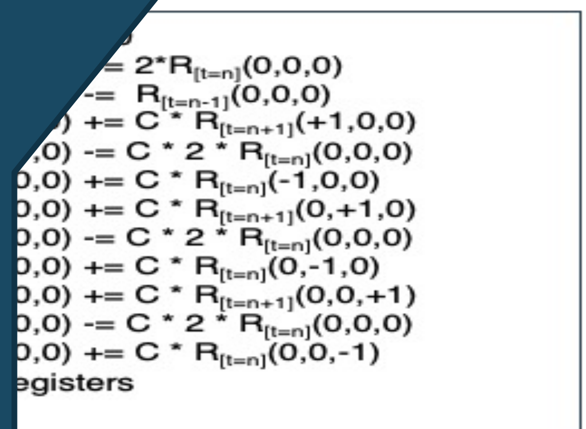
(must redesign for static dataflow and deep flow-through pipelines)

Von Neumann CPU



Dataflow (FPGA, GraphCore etc.)





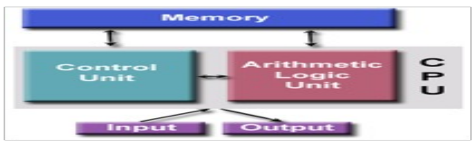
FPGA (*Field Programmable Gate Array):* Granularity of these operations and wires are single bits

CGRA *(Coarse Grain Reconfigurable Array)*: Programmability & ALUs at word granularity *improves speed and density!!* *(Cerebras, GraphCore, SambaNova, LPU)*

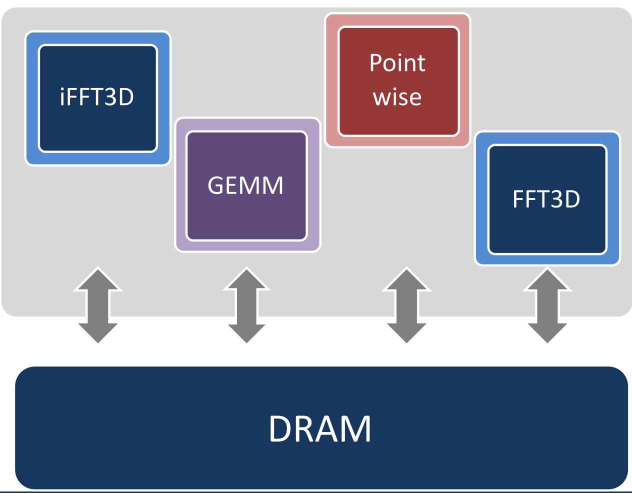
ASIC or Chiplet (*custom circuit):* Another factor of 10x on density and energy efficiency.

Algorithm Reformulated as Custom Circuit

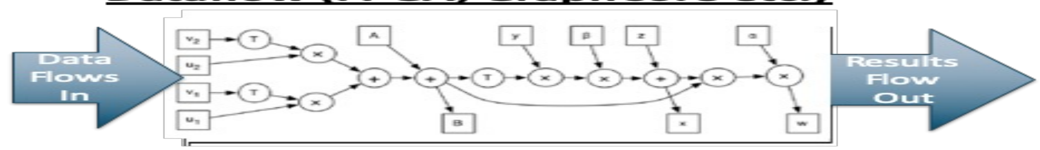
Von Neumann CPU



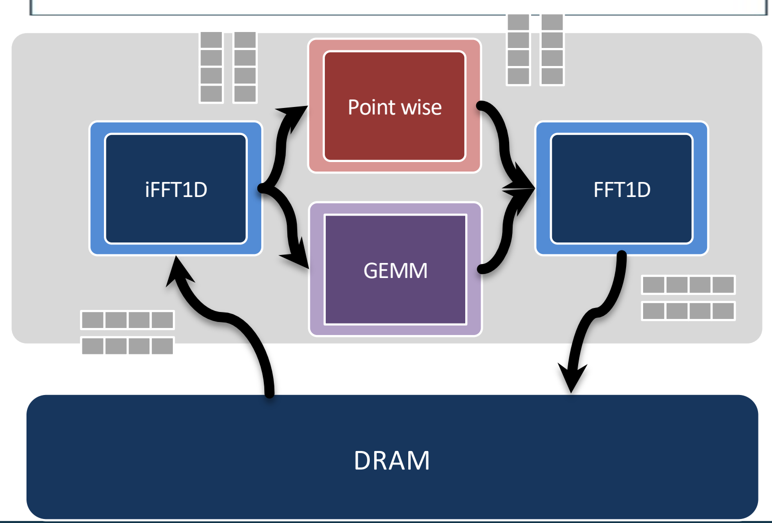

```
int main()
{
    int n = 0;
    while(n < 100)
    {
        n = n + 5;
        print("n = %d\n", n);
        pause(200);
        if(n == 50) break;
    }
    print("All done!");
}
```



Dataflow (FPGA, GraphCore etc.)



$R_{[t=n+1]}(0,0,0) = 0$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= 2 * R_{[t=n]}(0,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{-}= R_{[t=n-1]}(0,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= C * R_{[t=n+1]}(+1,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{-}= C * 2 * R_{[t=n]}(0,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= C * R_{[t=n]}(-1,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= C * R_{[t=n+1]}(0,+1,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{-}= C * 2 * R_{[t=n]}(0,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= C * R_{[t=n]}(0,-1,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= C * R_{[t=n+1]}(0,0,+1)$
$R_{[t=n+1]}(0,0,0) \mathrel{-}= C * 2 * R_{[t=n]}(0,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= C * R_{[t=n]}(0,0,-1)$
Rotate Registers



Preliminary Performance on CGRA HΨ

Eigenvalue Problem

$$h(i,j) = \langle \psi_i | H | \psi_j \rangle$$

Hpsi

$$P_i = H\psi_i - \epsilon_i \psi_i$$

$$P_i = A\left(P_i - \frac{\lambda_i}{\lambda_i^0} P_i^0\right)$$

Projection

$$P_i = P_i - \sum_{j=1,i} \langle P_i | \psi_j \rangle$$

$$\psi_i = \psi_i \cos\theta_i + P_i \sin\theta_i$$

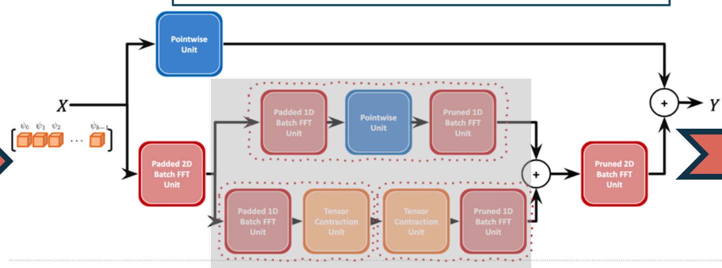
Orthogonalization

$$\psi_i = \psi_i - \sum_{j<i} \langle \psi_i | \psi_j \rangle$$

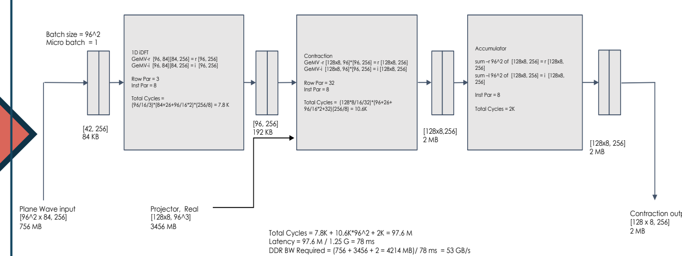
$$h(i,j) = \langle \psi_i | H | \psi_j \rangle$$

nline

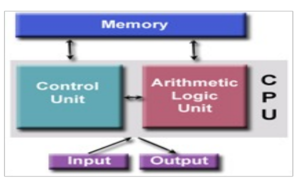
Dataflow Algorithm Reformulation



Mapping onto Custom Hardware

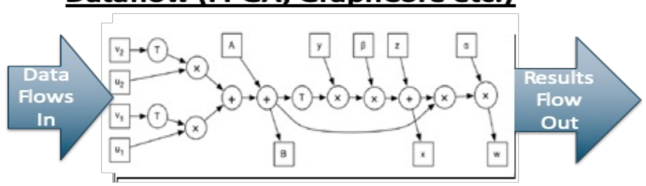


Von Neumann CPU or GPU




```
int main()
{
  int n = 0;
  while(n < 100)
  {
    n = n + 5;
    print("n = %d\n", n);
    pause(200);
    if(n == 50) break;
  }
  print("All done!");
}
```

Dataflow (FPGA, GraphCore etc.)

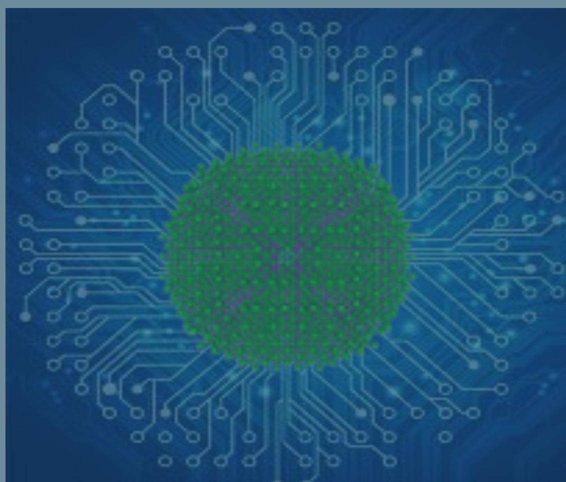


$R_{[t=n+1]}(0,0,0) = 0$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= 2^*R_{[t=n]}(0,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{-}= R_{[t=n-1]}(0,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= C * R_{[t=n+1]}(+1,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{-}= C * 2 * R_{[t=n]}(0,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= C * R_{[t=n]}(-1,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= C * R_{[t=n+1]}(0,+1,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{-}= C * 2 * R_{[t=n]}(0,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= C * R_{[t=n]}(0,-1,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= C * R_{[t=n+1]}(0,0,+1)$
$R_{[t=n+1]}(0,0,0) \mathrel{-}= C * 2 * R_{[t=n]}(0,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= C * R_{[t=n]}(0,0,-1)$
Rotate Registers

Accelerate the design of full custom accelerators!!

BERKELEY LAB

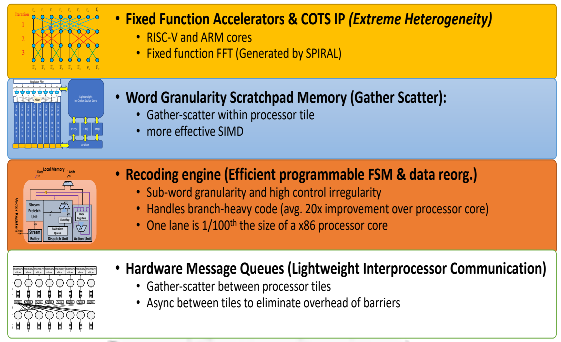
Thom Popovici, Andrew Canning (FFTx), Zhengji Zhang (NERSC)
Franz Francetti (CMU/FFTx)

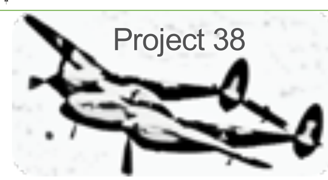


Specialization

Purpose built machines for big science targets.

Example: Google TPU. For DOE, DFT is 25% of workload

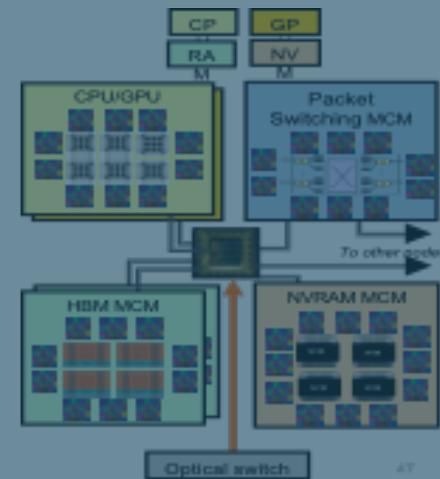




Heterogeneous Integration

Co-integration of many heterogeneous accelerators

Example: Apple Bionic chip, AWS Graviton2, Project38.



Resource Disaggregation

Photonic MCMs to enable reconfigurable nodes/systems

Example: Facebook/Google. Just DRAM utilization diversity in DOE could benefit from this.

Project 38 -- Background

DOD and DOE recognize the imperative to develop new mechanisms for engagement with the vendor community, particularly on architectural innovations with strategic value to USG HPC.

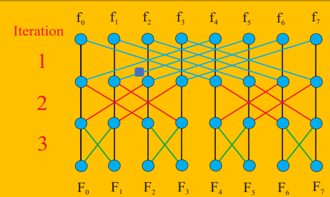
Project 38 (P38) is a set of vendor-agnostic architectural explorations involving DOD, the DOE Office of Science, and NNSA (these latter two organizations are referred to in this document as "DOE"). These explorations should accomplish the following:

- **Near-term goal:** *Quantify the performance value and identify the potential costs of specific architectural concepts against a limited set of applications of interest to both the DOE and DOD.*
- **Long-term goal:** *Develop an enduring capability for DOE and DOD to jointly explore architectural innovations and quantify their value.*
- **Stretch goal:** *Specification of a shared, purpose built architecture to drive future DOE-DOD collaborations and investments. (purpose-built HPC by 2025)*

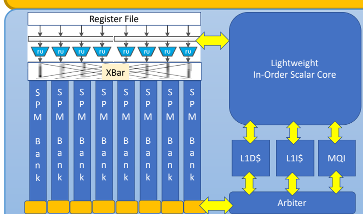
Recapping Key P38 Technology Features

innovative USG

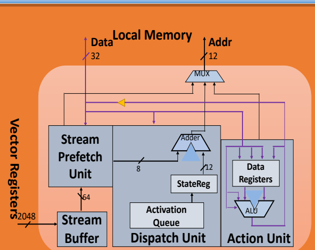
BERKELEY LAB
Lawrence Berkeley National Laboratory



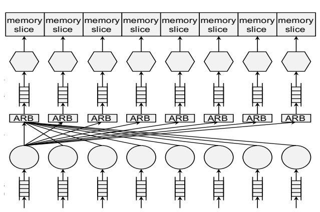
- **Fixed Function Accelerators & COTS IP *(Extreme Heterogeneity)***
  - RISC-V and ARM cores
  - Fixed function FFT (Generated by SPIRAL)



- **Word Granularity Scratchpad Memory (Gather Scatter):**
  - Gather-scatter within processor tile
  - more effective SIMD



- **Recoding engine (Efficient programmable FSM & data reorg.)**
  - Sub-word granularity and high control irregularity
  - Handles branch-heavy code (avg. 20x improvement over processor core)
  - One lane is 1/100th the size of a x86 processor core



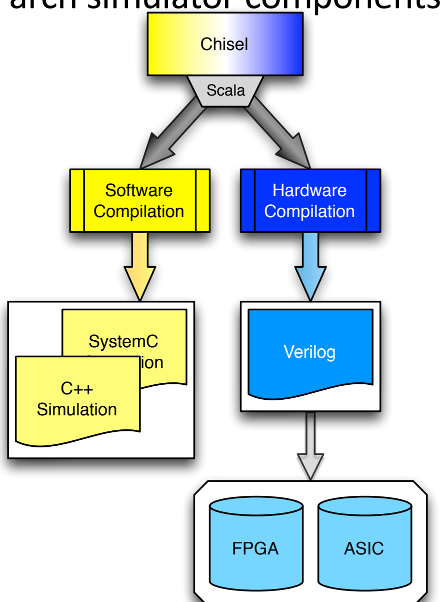
- **Hardware Message Queues (Lightweight Interprocessor Communication)**
  - Gather-scatter between processor tiles
  - Async between tiles to eliminate overhead of barriers

Hardware Generators: *Enabling Technology for Exploring Design Space Together with Close Collaborations with Applied Math & Applications*

Co-Develop Hardware and Algorithm

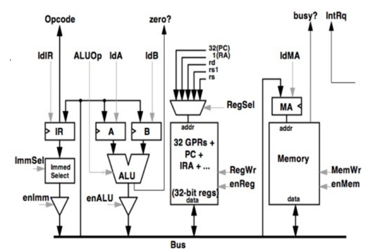
Chisel

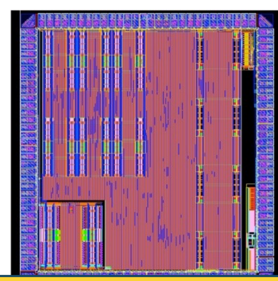
DSL for rapid prototyping of circuits, systems, and arch simulator components

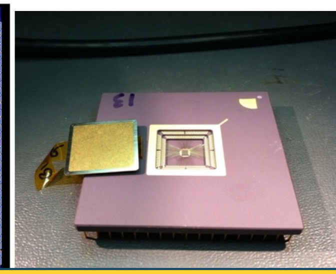


RISC-V

Open Source Extensible ISA/Cores

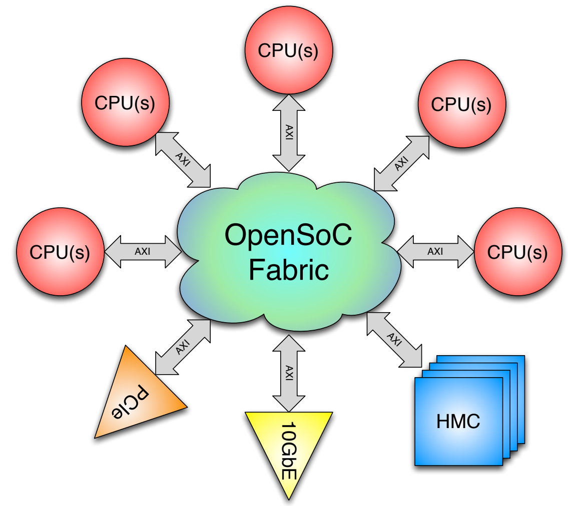






OpenSOC

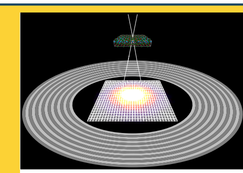
Open Source fabric To integrate accelerators And logic into SOC



SuperTools Superconducting RISC-V

QUASAR Quantum ISA

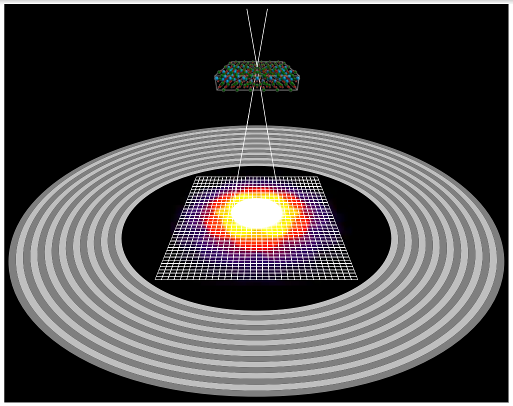
Multiagency Architecture Exploration



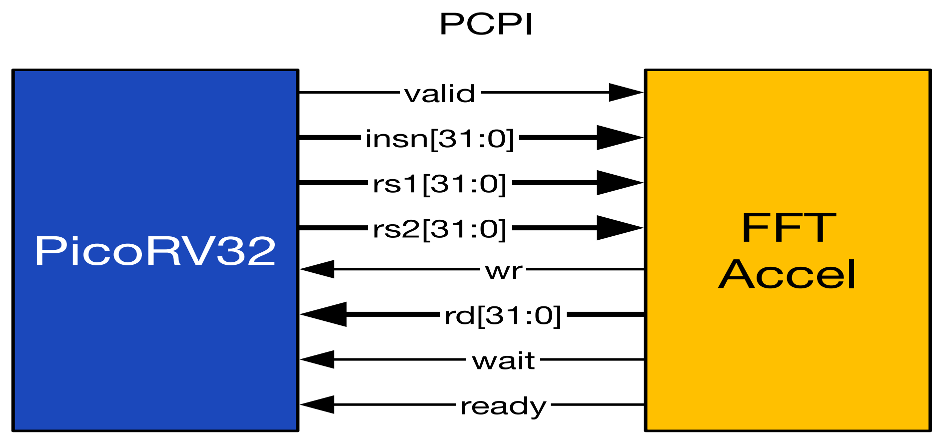
Active Sensors

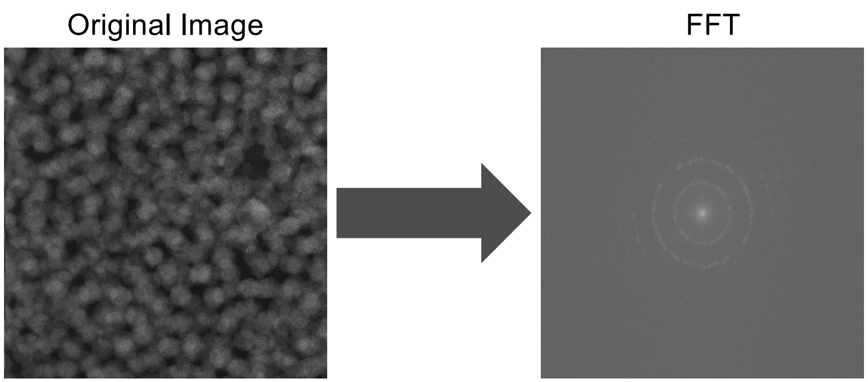
Results for RISC-V FFT Accelerator for CryoEM

Benchmarking FFT Accelerator for image analysis ***(Donofrio, Fard)***



Instruction	opcode[3:2]	Description
`fft_config`	10b	Configures FFT parameters
`fft_status`	01b	Reads FFTAccel status registers
`fft_start`	11b	Starts FFT processing
`fft_stop`	00b	Stops FFT processing





Created RISC-V Core with FFT ISA Extension

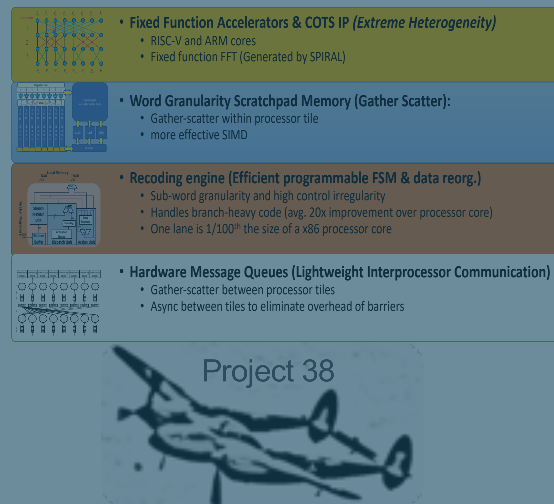
RISC-V+FFT Accel 126x faster than x86 host

- FFT on Intel Core i7-5930K @ 3.50GHz: **~265ms**
- FFTAccel (Floating): **~2.10ms**

Specialization

Purpose built machines for big science targets.

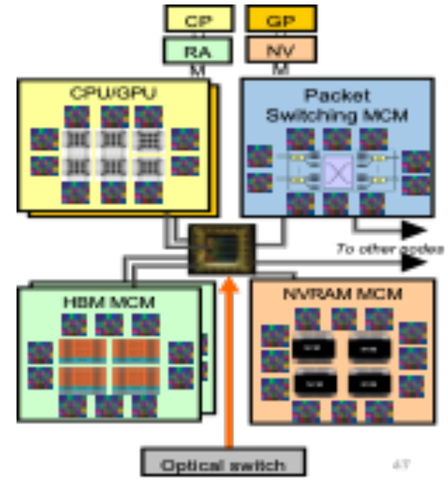
Example: Google TPU. For DOE, DFT is 25% of workload



Heterogeneous Integration

Co-integration of many heterogeneous accelerators

Example: Apple Bionic chip, AWS Graviton2, Project38.



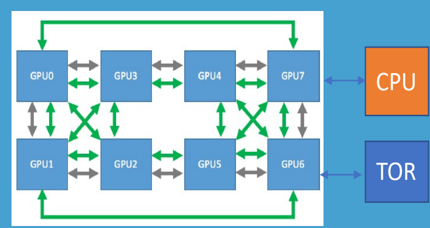
Resource Disaggregation

Photonic MCMs to enable reconfigurable nodes/systems

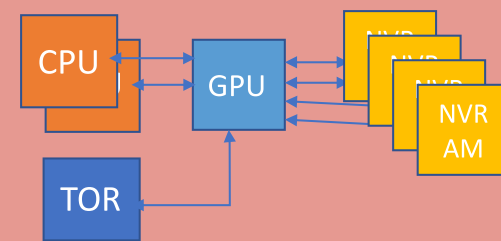
Example: Facebook/Google. Just DRAM utilization diversity in DOE could benefit from this.

Diverse Node Configurations for Datacenter Workloads

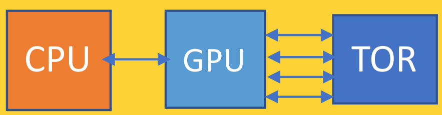
Training
- 8 connections: GPU
- 8 links to HBM (weights)
- 8 links: to NVRAM
- 1 links: to CPU (control)



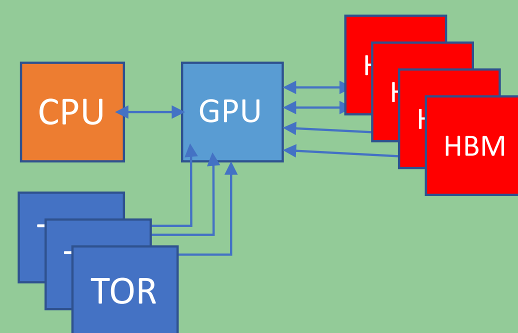
Data Mining
- 6-links: HBM
- 15 links: NVRAM (capacity)
- 4 links: CPU (branchy code)



Inference
- 16 links to TOR (streaming data)
- 8 links HBM (weights)
- 1 link: CPU

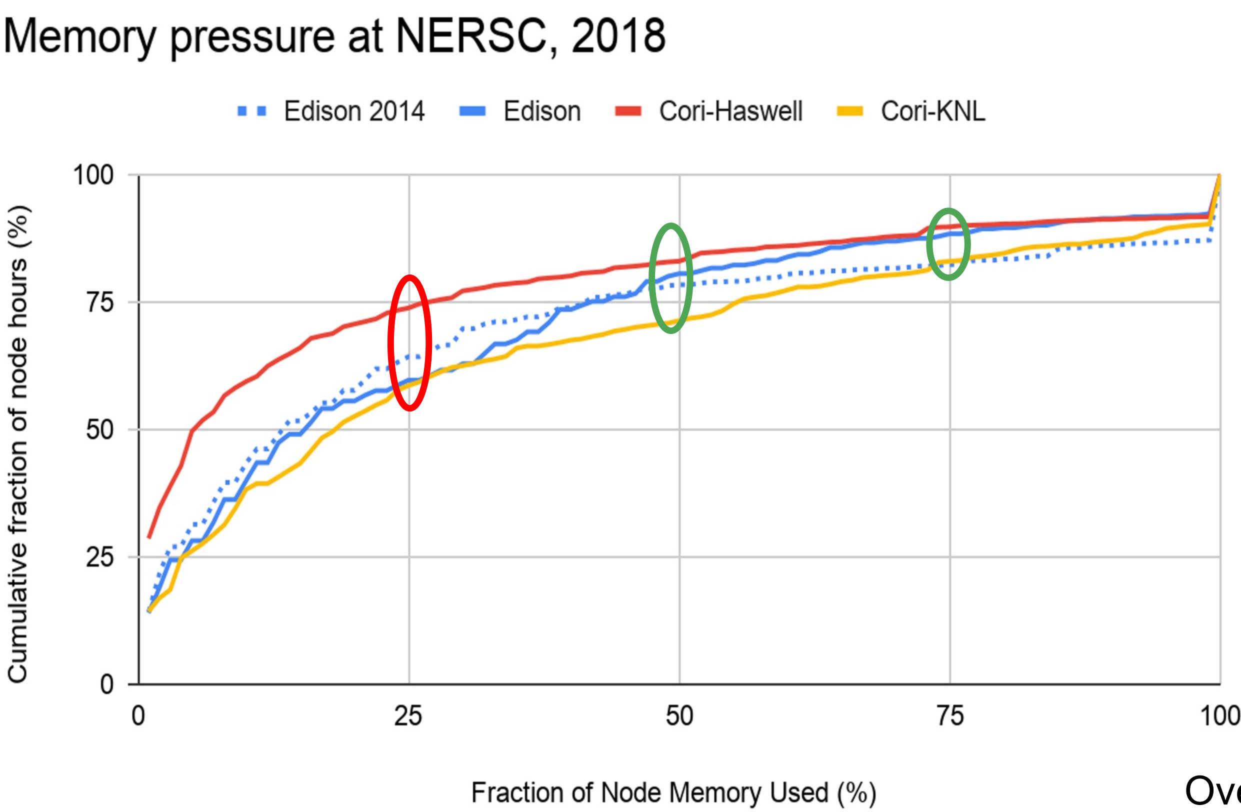


Graph Analytics
- 16 links HBM
- 8 links TOR
- 1 Link CPU



TOR NVRAM GPU HBM CPU

Memory Disaggregation



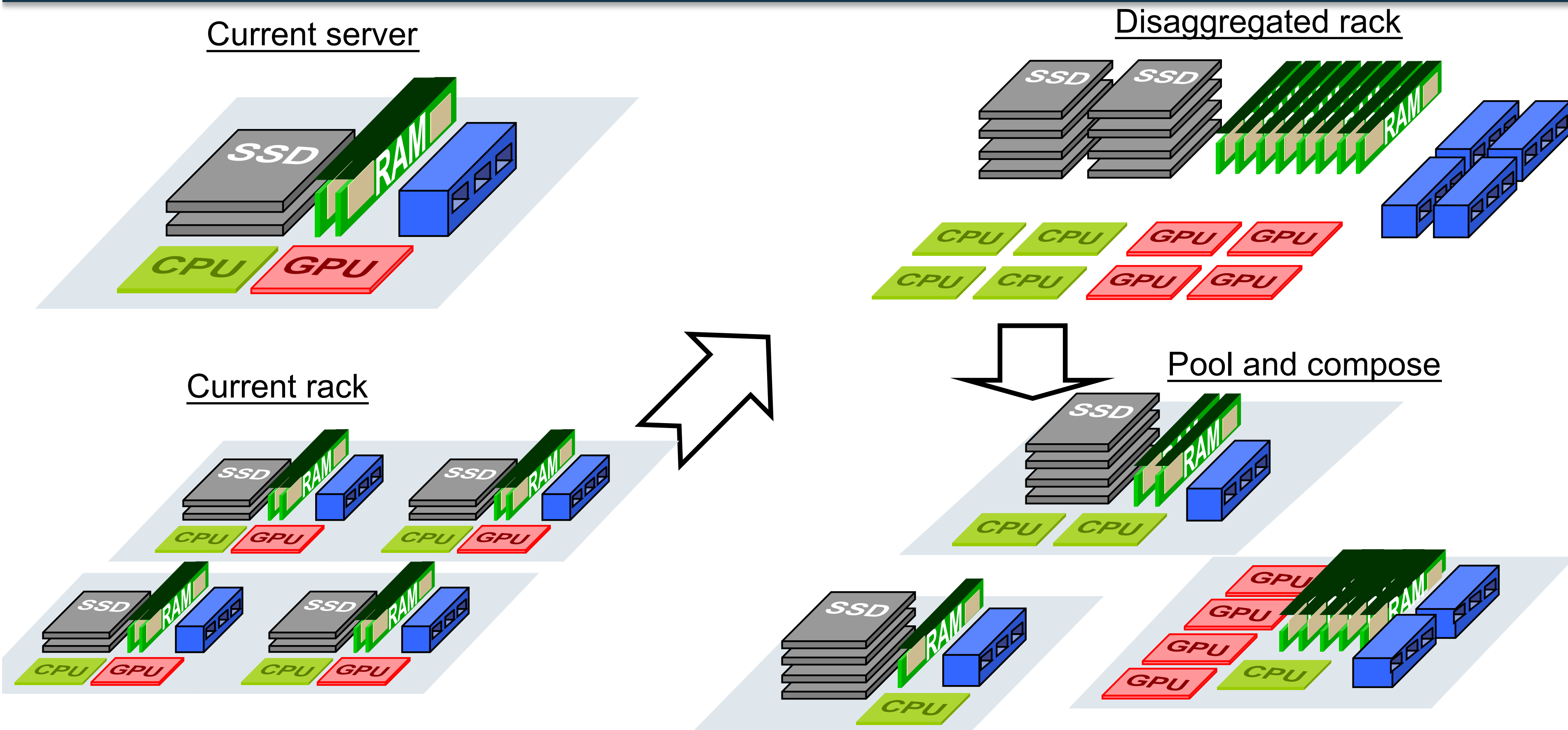
About 15% of NERSC workload uses more than 75% of the available memory per node.

And ~25% uses more than 50% of available memory.

But 75% of Haswell job hours (60% of KNL) use < 25% memory

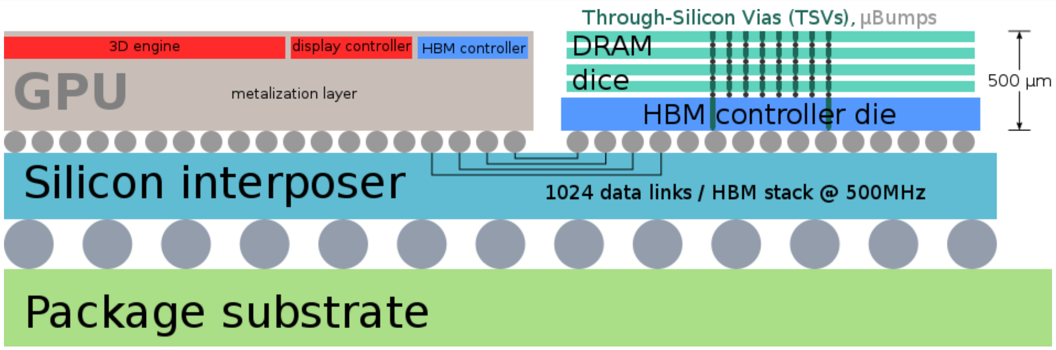
Overestimate: maxrss x ranks_per_node
Assumes memory balance across MPI ranks.

Disaggregated Node/Rack Architecture

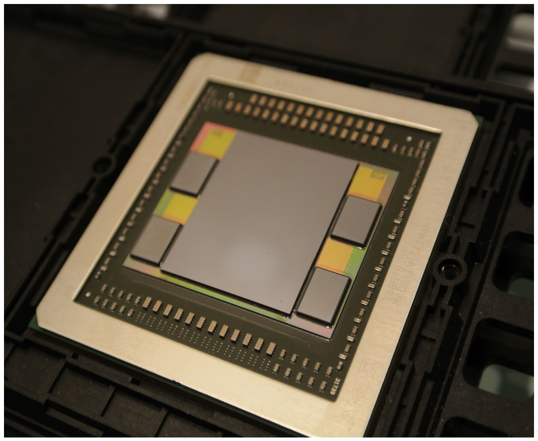


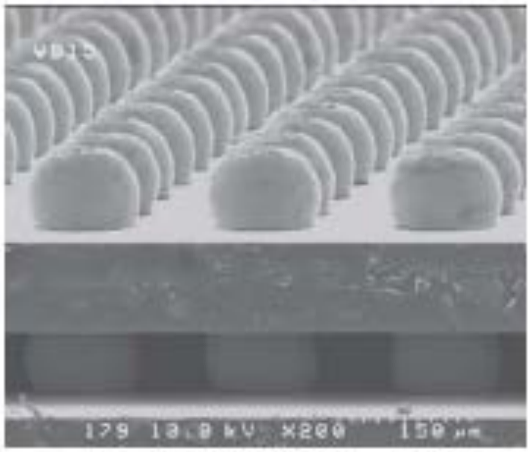
Most solutions current disaggregation solutions use Interconnect bandwidth (1 – 10 GB/s)
But this is significantly inferior to RAM bandwidth (100 GB/s – 1 TB/s)

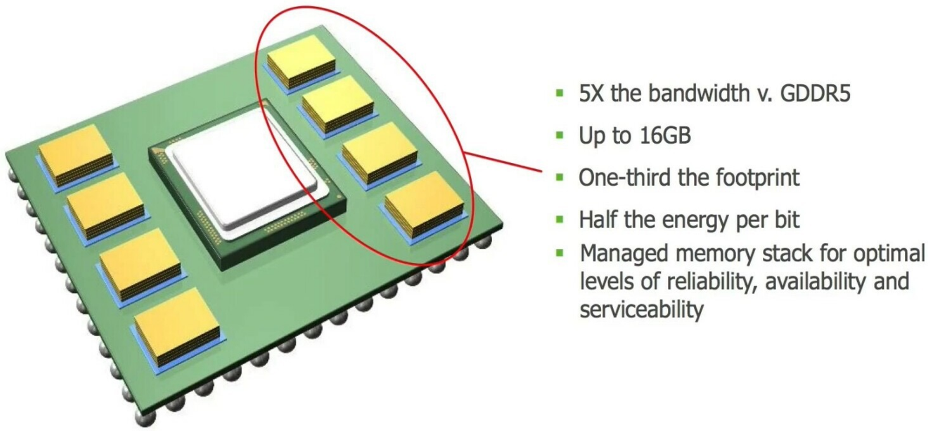
Interposers are the right point of intersection where copper pin bandwidth density could match photonics bandwidth density!



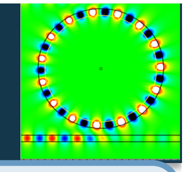
- **Good News:** Extend Bandwidth Density and lower power/bit
- **Bad News:** Limited (~2cm) reach
  - Cannot get outside of the package *(but we need to!!!!)*

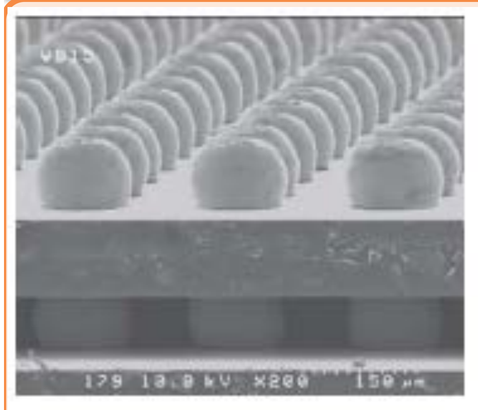


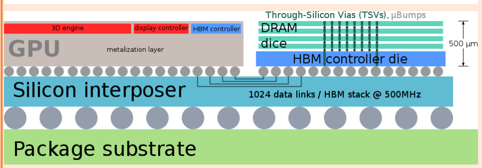




Impedance Matching to our Packaging Technology



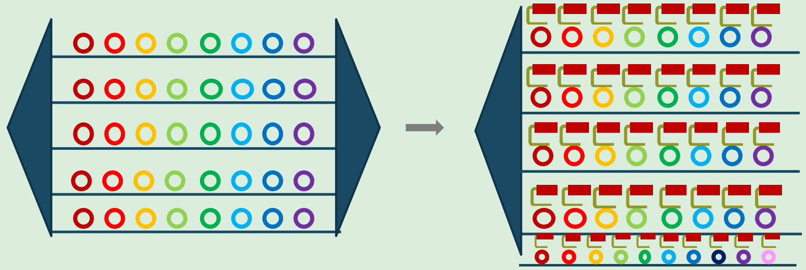


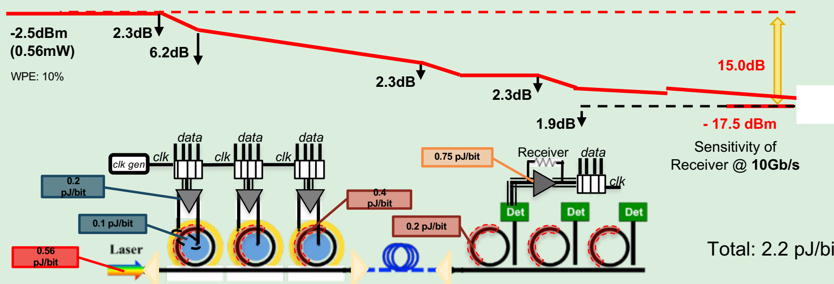


In-package integration

Solder Microbumps
& Copper Pillars @ 10Gbps

Wide and Slow!

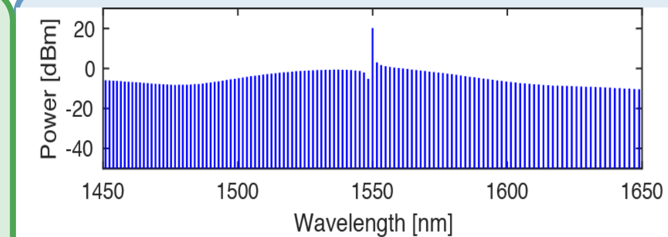


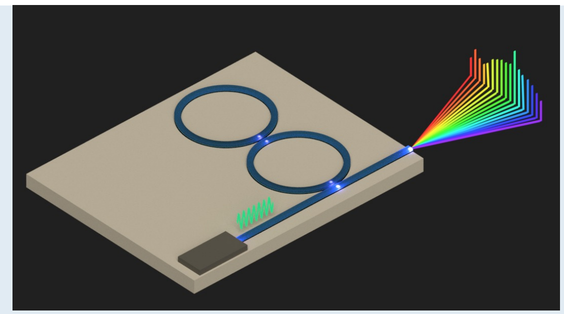


DWDM Using Silicon Photonics

Ring Resonators @ 10 Gigabits/sec per chan
Many channels to get bandwidth density

Wide and Slow!



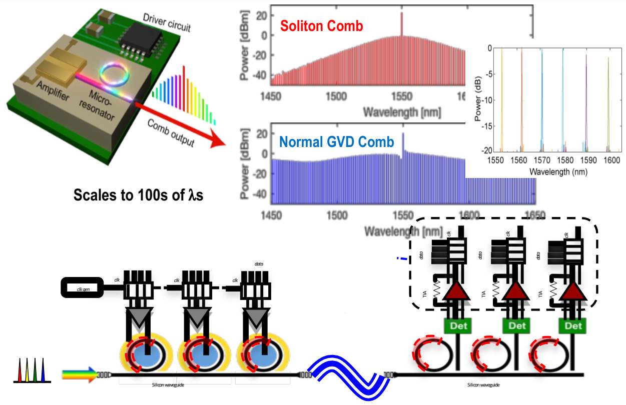


Comb Laser Sources

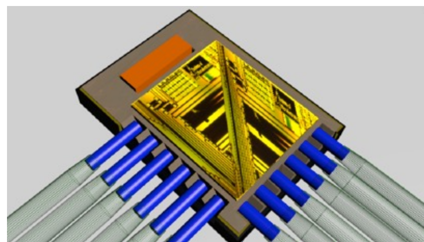
Single laser to efficiently
generate 100s of frequencies

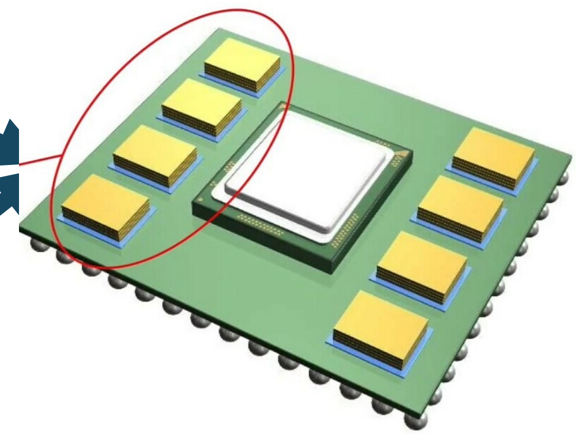
Wide and Slow!

Photonic MCM (Multi-Chip Module)

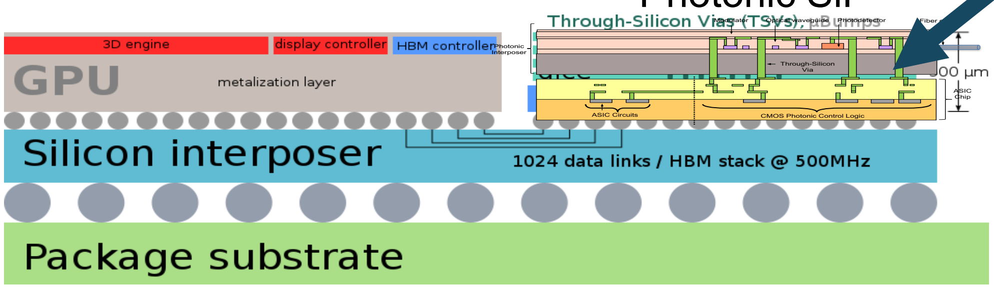


Comb Laser Source with DWDM Silicon Photonics
Wide-and Slow for high speed links

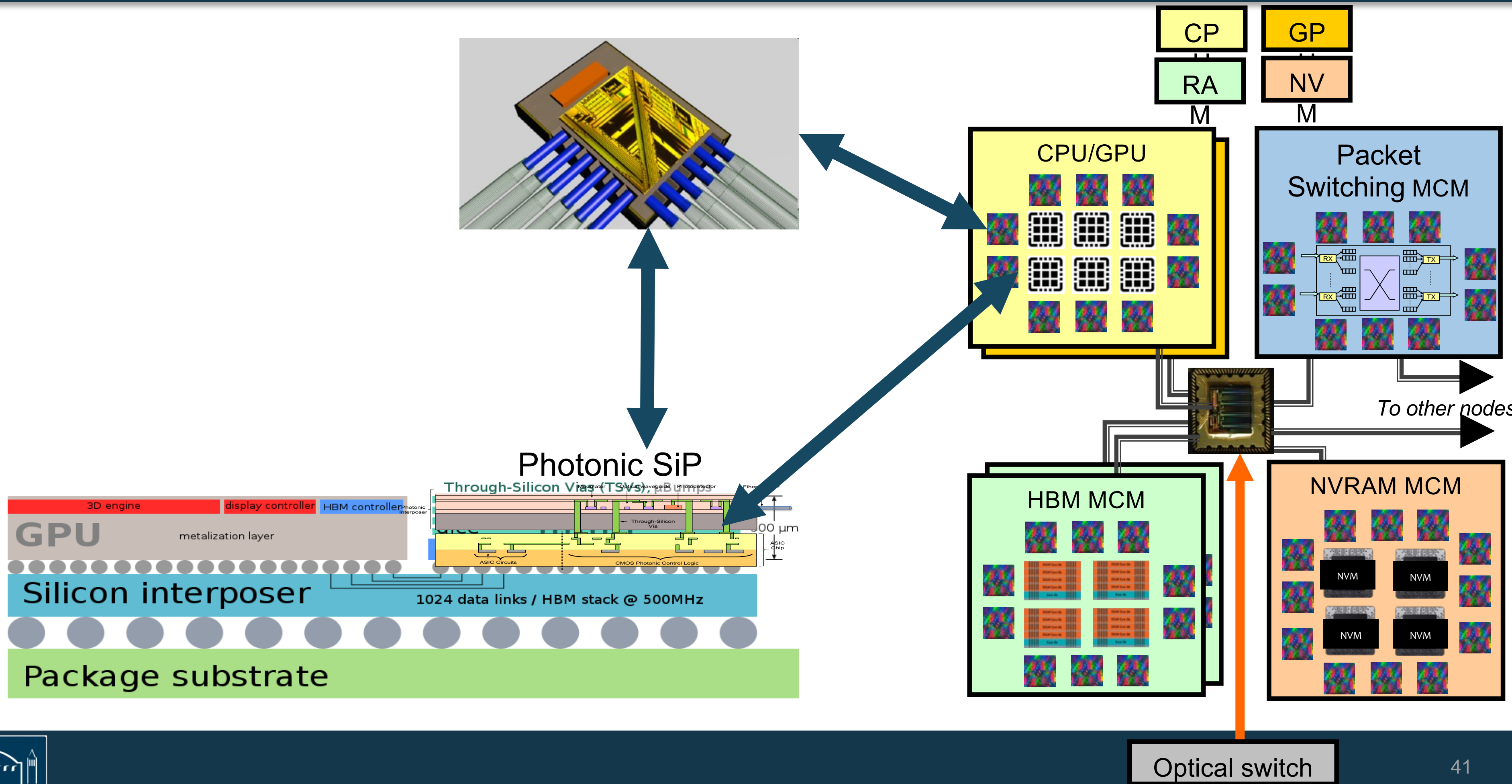




Photonic SiP

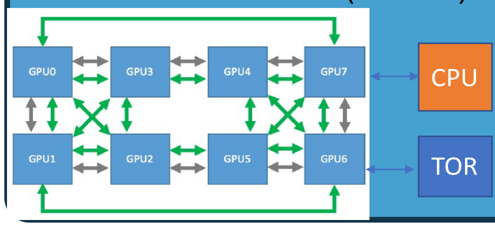


Photonic MCM (Multi-Chip Module)



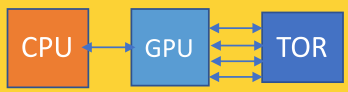
Training

- 8 connections: Peer GPU
- 8 links to HBM (weights)
- 8 links: to NVRAM
- 1 links: to CPU (control)



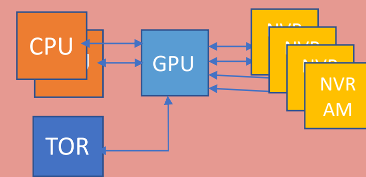
Inference

- 16 links to TOR (streaming data)
- 8 links HBM (weights)
- 1 link: CPU



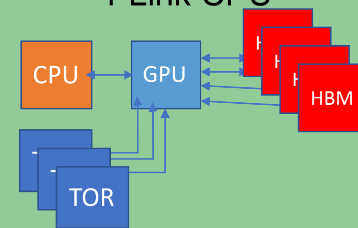
Data Mining

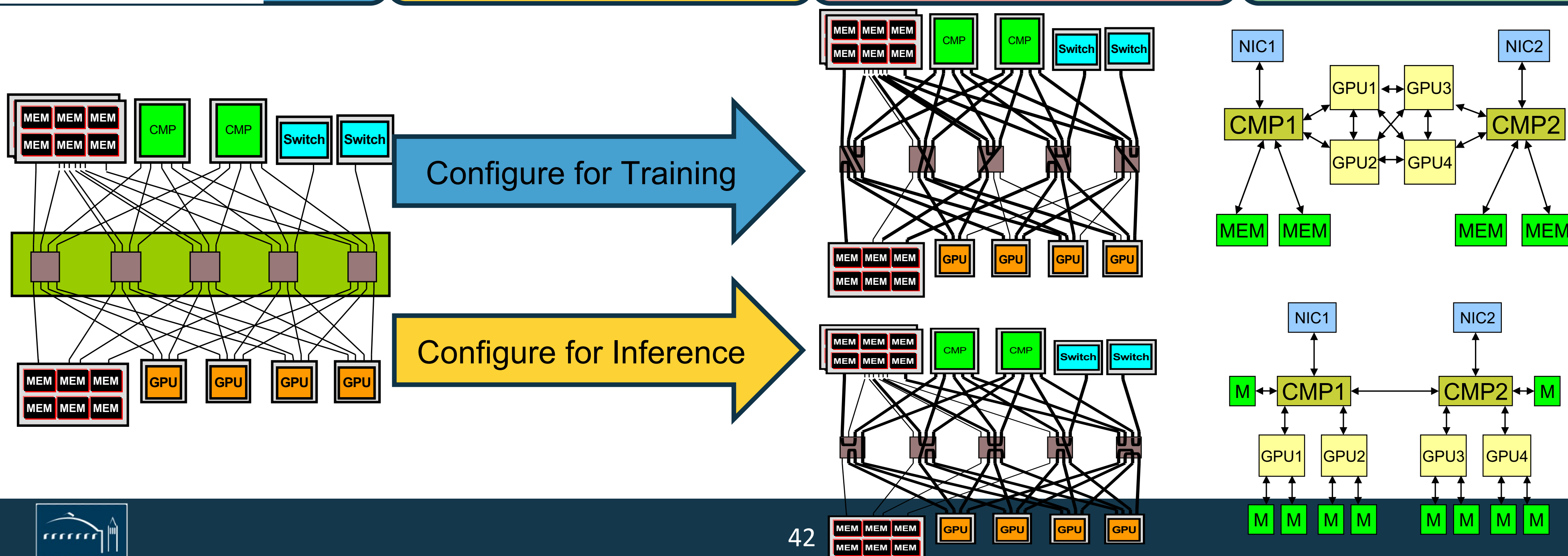
- 6-links: HBM
- 15 links: NVRAM (capacity)
- 4 links: CPU (branchy code)



Graph Analytics

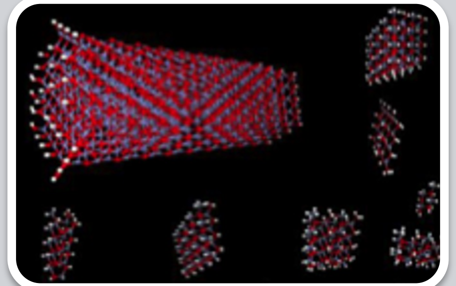
- 16 links HBM
- 8 links TOR
- 1 Link CPU

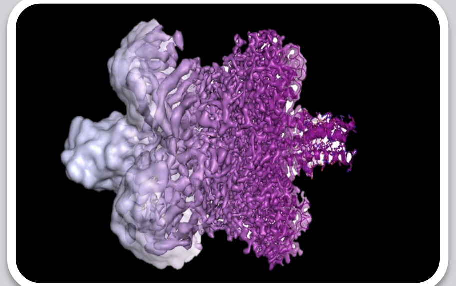


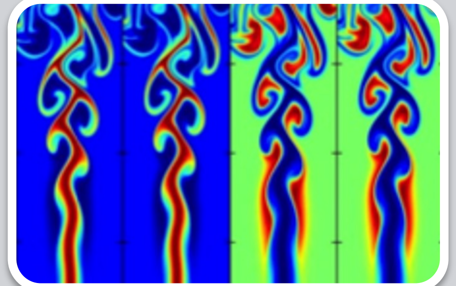


Architecture Specialization for Science

(hardware is design around the algorithms) can't design effective hardware without math







Materials

Density Functional Theory (DFT)

Use O(n) algorithm

Dominated by FFTs

FPGA or ASIC

CryoEM Accelerator

LBNL detector

750 GB / sec

Custom ASIC near detector

Genomics Accelerator

String matching

Hashing

2-8bit (ACTG)

FPGA solution

Digital fluid Accelerator

3D integration

Petascale *chip*

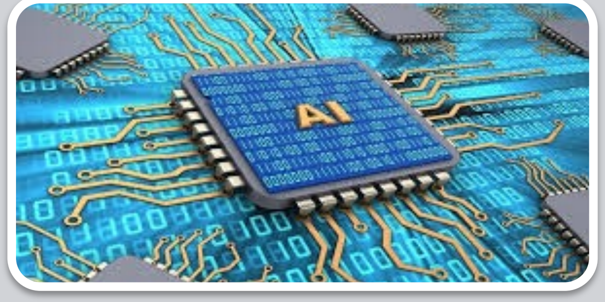
1024-layers

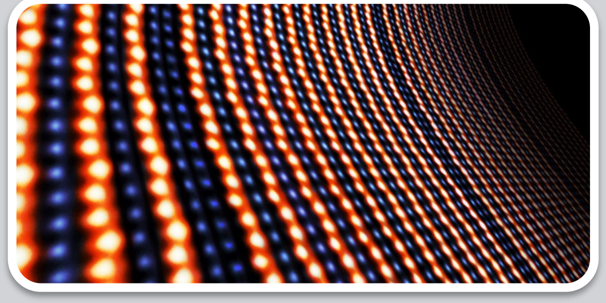
General / special HPC solution

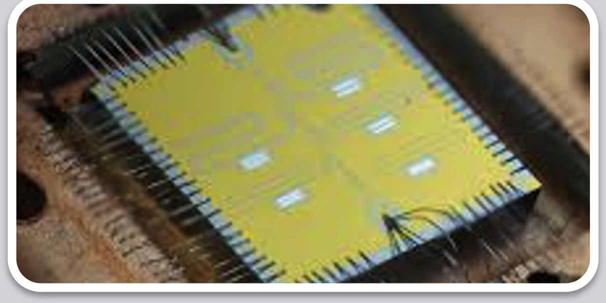
Conclusions

- **Think more seriously about how to put specialization productively to use for science**
  - Requires deep understanding of applied mathematics and the underlying algorithms to be successful
- **Reevaluate the business/economic model for the design and acquisition of HPC systems**
- **Accelerate the development of materials, devices, and systems for post-CMOS electronics**

Beyond-Moore Computing Directions







Heterogeneous Architectures

Specialized accelerators for performance / energy

Post CMOS Devices/Materials

Evaluate new devices using simulation across scales

New Models of Computation

Quantum algorithms, tools and testbeds, for science applications

Workload Analysis, Testbeds, Deployment